Achieving Private, Scalable, and Precise Data Collection in Wireless Sensor Networks

Saiyu Qi†‡, Zhenjiang Li† and Yunhao Liu‡

†Department of Computer Science and Engineering, Hong Kong University of Science and Technology, Hong Kong
‡School of Computer Engineering, Nanyang Technological University, Singapore

Abstract—Wireless Sensor Networks (WSN) become increasingly popular to collect data over a large area. Given the collected data set, the network manager can extract various kinds of aggregate statistics from the set to characterize the physical space. On the collection of the data, three requirements should be imposed: (1) Privacy: as sensor nodes are source limited and often deployed in an open environment, the sensed data suffer from privacy vulnerabilities. Secure mechanism should be provided to protect data privacy; (2) Communication efficiency: collecting data from large-scale sensor networks often involves large-volume data generation and transmission, which may quickly consume the energy of the WSN. To prolong the lifetimes of the sensor nodes, the sensed data should be transmitted in lightweight manner; (3) Accuracy: the sensed data should be recovered accurately at the base station (BS) so that the manager can manipulate them freely to achieve any precise aggregate statistic he prefers. To satisfy these requirements, we propose two novel privacy-preserving data collection schemes based on compressive sensing techniques. Our schemes address the privacy, communication efficiency and accuracy issues simultaneously. Detailed theoretical analysis and simulation results confirm the high performance of the proposed schemes.

Keywords—Data collection; Wireless Sensor Networks; Privacy

I. INTRODUCTION

Wireless sensor networks (WSN) are becoming increasingly popular for a wide range of applications. Wireless sensor nodes are deployed in various kinds of physical areas such as factories, office areas, forests or rivers to sense the physical space [15], [18], [19]. Sensor nodes convert physical parameters such as heat, light intensity or temperature to data items and send them to base station (BS) through multiple intermediate nodes.

In many sensor network applications, the network manager is interested in various kinds of aggregate statistics (such as SUM, AVERAGE, MAX/MIN, MEDIAN, HISTOGRAM, PERCENTILE, TOP-k, TRENDLINE) from the sensed data over a physical area, which are often used to present the physical characteristics of the area. To achieve these statistics, the base station needs to collect the sensed data from the whole area. However, successful accomplishment of data collection suffers privacy issues. As sensor nodes are often deployed in open environments, it is easy for an adversary to eavesdrop on the communication links to explore the content of the data. Even worse, sensors cannot provide temper-resistance for cost consideration. An adversary can thus easily compromise them to obtain their sensed data and control them.

To eliminate such privacy vulnerabilities, several privacy-preserving schemes have been proposed. These schemes, however, only aim to design secure mechanism to protect data privacy without the consideration of communication efficiency or accuracy, both of which are also essential in the data collection process. In particular, (1) For communication efficiency: as sensors are usually resource-limited and power-constrained [17], forwarding large volume data in-network may quickly consume the energy of the WSN since data transmission is a highly energy-consuming operation. To maximize sensor lifetime, it is essential to improve communication efficiency during the data collection process. (2) For accuracy: during the data collection process, each node will generate a data and forward it to the BS. Each of the data should be recovered individually and precisely by the BS so that it can manipulate them freely to achieve precise aggregate statistic. Most of the previous privacy-preserving schemes suffer from the drawback of either low communication efficiency or inaccuracy, as we will discuss in detail in Section II.

In this paper, we systematically study the problem of private, scalable and precise data collection in wireless sensor networks. We design two privacy-preserving schemes to solve this problem. Firstly, our schemes can guarantee both communication efficiency and accuracy. To achieve this goal, we propose to use a novel technique called compressive sensing (CS) [7-9], which shows that spatially correlated data can be compressed during the transformation and precisely recovered when needed, as the data forwarding infrastructure for both of our two schemes. Such a utility is based on the observation of the inherent nature of the sensed data. As WSN are in general deployed in a specific area, data are not generated independently at each individual node. Instead, there exists spatial correlation between them. Such a correlation provides us a potential to compress these
data by using compressive sensing (CS) technique during the forwarding process.

Secondly, our schemes provide data privacy to resist both eavesdropping attack and compromising attack with moderate computational overhead. To achieve this goal, we integrate the CS based data forwarding infrastructure with several lightweight cryptographic primitives. We design our first scheme by integrating two cryptographic primitives: pseudorandom permutations (PRP) [10] and symmetric encryption with the CS based data forwarding infrastructure. The advantage of this scheme is that it partially relies on the nature of CS technique to provide data privacy, and thus introduce little computational overhead. This scheme is resilient to the eavesdropping adversary and allows the compromising adversary to learn only restricted knowledge about the sensed data. To further eliminate the threat of compromising attack, we design the second scheme by integrating pseudorandom permutations (PRP) [10] and additively homomorphic encryption (AHE) [4] with the CS based data forwarding infrastructure. The basic idea is that we do not rely on the CS technique itself to protect data privacy. Instead, the sensed data are locally encrypted at each sensor node before sending out. We choose additively homomorphic encryption as our encryption mechanism so that the encrypted data can still be compressed in-network. Such a construction largely enhances the data privacy as the adversary cannot obtain the sensed data unless it captures the encryption key. Comparing with the first scheme, the second scheme provides a strict cryptography guarantee against the compromising threat.

The rest of this paper is organized as follows. Section II discusses related works. We introduce technique preliminaries in Section III. We describe the network and adversary models in Section IV. We present the design of our schemes in section V. We conduct security analysis and performance evaluation in Sections VI and VII respectively.

### II. RELATED WORK

Previous private-preserving schemes often raise high communication overhead or can only collect inaccurate data. They can be classified into two categories.

**Basic encryption scheme:** In basic encryption scheme, each sensor shares a symmetric key with the BS for data encryption. After forwarding over multiple hops, the encrypted data will be decrypted at the BS. As each data is forwarded individually and under encryption, the requirements of privacy and accuracy can be satisfied. However, this scheme suffers from poor communication efficiency which consumes the energy of sensors in two folds: (1) Overall energy consumption: as each sensed data needs to be forwarded through multi-hop routing from the individual sensor nodes to BS, transmitting large number of them will quickly consume the energy of the sensors. (2) Energy balance: as the amount of data forwarded by those sensors closer to the BS is up to several orders of magnitude larger than that of the nodes on the boundary of sensing area, energy consumption is not well balanced. As a result, sensors close to BS will quickly consume their energy and lose functionality.

**Private data aggregation schemes:** To overcome the above efficiency issues, private data aggregation schemes [1-6] are proposed. Through data aggregation, the overhead of transformation is dramatically decreased. However, data accuracy is also degrade in the aggregation process as BS can only get an inaccurate aggregation result, such as the summation of the sensed data, from which it is hard to extract various sophisticated aggregate statistics. In particular, the summation can only be directly used to extract limited aggregate statistics like COUNT and AVERAGE. More advanced aggregate statistics such as HISTOGRAM, PERCENTILE, TOP-k, TRENDLINE are hard to obtain. The private data aggregation schemes can be divided into several types based on the aggregation manner, among which the most efficient type is end to end manner [3], [4], [6]. Comparing with other types, end to end manner consumes less energy and achieves better energy balance. For this reason, we will only consider end to end manner in this study. By using an end to end private data aggregation scheme, sensed data are aggregated at intermediate nodes. Each intermediate node forwards an aggregation of data, which is generated from the sensed data of all downstream nodes. To protect data privacy, all the data are encrypted in the forwarding process. In the remainder of this paper, we just term end to end private data aggregation as private data aggregation for simplicity.

### Table I

**FREQUENTLY USED NOTATIONS**

<table>
<thead>
<tr>
<th>Notation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Q</td>
<td>The number of sub trees</td>
</tr>
<tr>
<td>u_j</td>
<td>In a sub tree, u_j denotes a single sensor node which has j hops to BS</td>
</tr>
<tr>
<td>d_j</td>
<td>A sensed data generated at u_j</td>
</tr>
<tr>
<td>d</td>
<td>A data vector consists of multiple d_j</td>
</tr>
<tr>
<td>h_{ij}</td>
<td>A homomorphic key of u_j</td>
</tr>
<tr>
<td>k_j</td>
<td>A PRP key of u_j</td>
</tr>
<tr>
<td>S_{ij+1}</td>
<td>A symmetric key shared between u_i and u_{i+1}</td>
</tr>
<tr>
<td>u_{q=1}^{N}</td>
<td>All the downstream nodes of u_j</td>
</tr>
<tr>
<td>Z_{j+1}</td>
<td>The size of {u_q}_{q=j+1}</td>
</tr>
<tr>
<td>M</td>
<td>The dimension number of the compressed vector</td>
</tr>
<tr>
<td>R</td>
<td>A random number generated at BS</td>
</tr>
<tr>
<td>S</td>
<td>A secure parameter generated at BS</td>
</tr>
<tr>
<td>A</td>
<td>A sensing matrix</td>
</tr>
<tr>
<td>a_{i,j}</td>
<td>An element of A</td>
</tr>
</tbody>
</table>
III. TECHNIQUE PRELIMINARIES

A. Introduction of compressive sensing

The compressive sensing (CS) technique [7-9] is a kind of compression technique for data transmission and sharing in pervasive platforms. The key idea of CS is that spatially correlated data can be jointly transmitted in an efficient compressed pattern rather than one by one. Consider a WSN consisting of $N$ nodes and a BS, each node having a sensed data $d_j$, $j=1,\ldots,N$, which forms a spatially correlated data vector $d = \{d_1, d_2, \ldots, d_N\}$. To get $d$, a trivial way is to require each node to report its sensed data one by one to BS. Instead, CS provides a more efficient way in which the vector $d$ can be accurately recovered from a small number $M$ ($M << N$) of linear combination vector $y = \{y_1, y_2, \ldots, y_M\}$. Each element $y_j$ is a linear combination of all the elements of $d$: $y_j = \sum_{i=1}^{N} a_i d_i$ where $a_i$ is a constant number. We can present $y$ as: $y = A \times d$. CS guarantees that, for a random sensing matrix $A$, $d$ can be accurately recovered from $y$.

The utilization of CS is based on the assumption that the sensed data has spatial correlation. Actually, work presented in [12], [13] is among the first to use CS to recover sensed data from WSN. Experiments on real data sets [13] show that sensed data indeed has spatial correlation and can be precisely recovered by using CS. Furthermore, through reorganization, even data without spatial correlation can be precisely recovered by using CS. The schemes [12], [13] however, cannot be directly applied in our target scenario since even an eavesdropping adversary can recover all the sensed data of WSN. Recently, [20], [21] propose to use compressive sensing as encryption to secure the transmitted signals. Unfortunately, their schemes still cannot defend eavesdropping attack if deployed in our scenario.

B. Cryptographic primitives

Additively homomorphic encryption (AHE): We choose [4] as the candidate of our additive homomorphic encryption, which can be formally described as follows: Let $HE()$ denote the encryption algorithm. Let $hk_i$ denotes a homomorphic key and $MS$ the domain of the message space. Given two ciphertexts $c_1 = HE_{hk_i}(m_1)$ and $c_2 = HE_{hk_2}(m_2)$, $(c_1 + c_2) \mod MS$ is equal to $HE_{hk_i}(m_1 + m_2)$ where $hk = hk_1 + hk_2$.

Pseudorandom permutation (PRP): A pseudorandom permutation function $f: H \times P \rightarrow P$ is a pseudorandom permutation if given a fixed PRP key $k_i$ belongs to $H$, the input-output behavior of a random instance of $f$ is "computationally indistinguishable" from that of a random permutation on a domain $P$. In this paper, we assume $P = \{0, 2^r\}$ where $r$ is a parameter determined at initialization.

IV. NETWORK AN ADVERSARY MODELS

A. System model

In this paper, we consider a wireless sensor network composed of a BS located at the center of the physical area and a large number of sensor nodes around it. Each sensor node monitors its direct environment and can generate sensed data. We assume that clocks on different sensors can be efficiently calibrated [16] and each sensed data is in the range $[0, U]$ [6]. The network manager can require BS to collect data.

In practice, sensor nodes are often organized as one or multiple long routes originated from BS and extended hop by hop to the boundary of the physical area [15]. In this study, we assume that all the nodes are organized as a collection tree which consists of a root BS and $Q$ sub trees. Each of the sub trees consists of $N$ sensor nodes that form a multi-hop route originated from BS. Given a sub tree, we denote its $N$ nodes as $u_1, u_2, \ldots, u_N$. For a node $u_j$, the index $j$ represents the hop numbers from BS to $u_j$. We use the set $\{u_q\}_{q=0}^{N}$ to present all the downstream nodes of $u_j$ and $Z_{j+1} = \sum_{q=0}^{N} 1$ to present the size of the set. Also, we call $u_N$ leaf node and the remainder nodes from $u_{N-1}$ to $u_1$ intermediate nodes. To collect data, BS first broadcasts a request command and the collection process is proceeded on all the sub trees concurrently. In each sub tree, data are transferred along the order: $u_1 \leftarrow u_2 \leftarrow \cdots \leftarrow u_N$. Figure 1 shows an example of a collection tree which has three sub trees, each containing three nodes. We assume that in a sub tree, each node $u_j$ knows its index $j$, direct parent and direct child.

B. Adversary model

We consider a general adversary with passive and active abilities to explore the data privacy of sensor networks. In particular, we consider:

- The adversary can eavesdrop on a communication link between two sensor nodes and try to extract useful data from it. In this study, we assume that the adversary has strong ability to launch eavesdropping attack, i.e., the adversary knows the topology of the collection tree and can adaptively select any links to eavesdropping.
- The adversary can compromise a subset of sensor nodes. After compromising several nodes, the adversary can im-
V. Scheme design

In this section, we describe the concrete design of our two privacy-preserving schemes. In both of our two schemes, each node shares a PRP key with BS and uses its PRP key to generate a data report from its sensed data. An intermediate node receives data reports from its downstream nodes, compresses them as well as its own data report to generate a compressed data report. On the other hand, as BS knows these PRP keys, it can recover the sensed data immediately.

Step 2. Query dissemination When BS wants to collect data, it first generates a random number $R$ which is different in each query. BS then broadcasts the query message $m=\{\text{query request}, R, M\}$ to the network. After broadcasting, each node records $m$. As the data collection is performed on sub tree basis, we consider the collection process in a single sub tree in the following.

Step 3. Data report at leaf node $u_N$ After receiving $m$, $u_N$ begins to send its data report as follows:

1. $u_N$ generates a sensed data $d_N$ and establishes a data report $r_N$ as follows: $u_N$ derives a set of random numbers $\{a_{qN}\}_{q=1}^M$ where $a_{qN}=f_{k_u}(R)$ and for $q = 1$ to $M-1$, $a_{q+1N} = f_{k_u}(a_{qN})$. Then, $u_N$ computes $r_N$ as: $r_N = \{a_{1N} d_N, a_{2N} d_N, \ldots, a_{MN} d_N\}$.

2. $u_N$ uses $sk_{N-1}$ to generate a symmetric encryption $\text{Enc}_{sk_{N-1}}(r_N)$ and transmits it to $u_{N-1}$.

Step 4. Data compression at each intermediate node $u_j (N-1 \leq j \leq 1)$ $u_j$ receives a message from its child $u_{j+1}$ and decrypts it using the shared private key. Suppose $u_j$ has obtained a compressed data report $r_{j+1}$ from its child $u_{j+1}$ when $u_{j+1}$ is $u_N$, it generates a new compressed data report $r_j$ as follows:

1. $u_j$ generates its own data report $r_j$ as follows: $u_j$ derives a set of random numbers $\{a_{qj}\}_{q=1}^M$ where $a_{1j}=f_{k_u}(R)$ and for $q=1$ to $M-1$, $a_{q+1j} = f_{k_u}(a_{qj})$. Then, $u_j$ sets $r_j$ as: $r_j = \{a_{1j} d_j, a_{2j} d_j, \ldots, a_{Mj} d_j\}$.

2. $u_j$ adds $r_j$ and $r_{j+1}$ to generate $r_j$ as: $r_j = \{\sum_{q=1}^N a_{1qj} d_q^a1j, \ldots, \sum_{q=1}^N a_{Mqj} d_q^aMj\}$.

3. $u_j$ uses $sk_{j-1}$ to generate a symmetric encryption $\text{Enc}_{sk_{j-1}}(r_j)$ and sends it to $u_{j-1}$.

Step 5. Data recovery Finally, BS decrypts the compressed data report $r_j$ from $u_1$. $r_j$ can be presented as:

$$r_j = \begin{pmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{M1} & \cdots & a_{MN} \end{pmatrix} \times \begin{pmatrix} r_1 \\ \vdots \\ r_N \end{pmatrix}$$

For each $u_i$, BS uses $k_i$ to derive $\{a_{qi}\}_{q=1}^M$. In this way, BS can get the full sensing matrix $A$. As each element $a_{qi}$ of $A$ is generated by the PRR function $f$, $A$ is a random matrix. BS can then recover the precise data vector $d=(d_1, d_2, \cdots, d_N)$ by using CS. BS does the same thing to recover $d$ for each sub tree. As a result, all the sensed data are recovered.

B. Our second scheme: privacy-enhanced data collection

In our second scheme, data reports are encrypted at each sensor node before sending out. Such a construction largely enhances the data privacy as the adversary cannot obtain the data reports unless it knows the encryption key.
Furthermore, to ensure that the encrypted data reports can still be compressed at intermediate nodes, we propose to use additively homomorphic encryption, which allows direct compression among ciphertexts without decryption.

**Step 1. Initialization** Prior to network deployment, BS sets up system parameters for each sub tree as follows:

1. BS generates a PRP key $k_i$ for each node $u_i$ and distributes $k_i$ to $u_i$. $k_i$ is used for $u_i$ to generate its data report.
2. BS generates a homomorphic encryption key $hk_i$ for each node $u_i$ and distributes $hk_i$ to $u_i$ for encryption. BS also securely saves each $hk_i$ in its database.
3. As BS knows the total number $N$ of sensor nodes in a subtree, it can generate dimension degree $M$ and other parameters of CS.

**Step 2. Query dissemination** BS generates a random number $R$ and broadcasts query message $m = \text{query request}, R, M$). Also, we consider the collection process in a single sub tree in the following.

**Step 3. Data report at leaf node $u_N$** When receiving $m$, $u_N$ begins to send its data report as follows:

1. $u_N$ generates a data $d_N$ and establishes $r_0 = \{a_1d_N, a_2d_N, \ldots, a_Md_N\}$ as the same as the first scheme.
2. $u_N$ then uses $hk_N$ to encrypt $r_0$ as an encrypted version: $ero = \{HE_{hk_N}(a_1d_N), HE_{hk_N}(a_2d_N), \ldots, HE_{hk_N}(a_Md_N)\}$ and forwards $ero$ to its parent $u_{N-1}$.

**Step 4. Data compression at intermediate node $u_j$ ($N-1 \leq j \leq 1$)** When $u_j$ receives an encrypted $ero_j$ of a compressed data report from its child $u_{j+1}$ (ero$_N'$ when $u_{j+1}$ is $u_N$), it generates a new encryption $ero_j$ of a compressed data report as follows:

1. $u_j$ generates its own data report $r_{1j} = \{a_1d_j, a_2d_j, \ldots, a_Md_j\}$.
   - $u_j$ then uses $hk_j$ to encrypt $r_{1j}$ as: $ero_j = \{HE_{hk_j}(a_1d_j), HE_{hk_j}(a_2d_j), \ldots, HE_{hk_j}(a_Md_j)\}$.
2. $u_j$ adds up $ero_{j+1}$ and $ero_j$ to generate $ero_j$ as:

   $ero_j = \{HE_{hk_j}(\sum_{q=j+1}^{N}a_qd_q) + HE_{hk_j}(a_1d_j)\}modM$.

   $\cdots, HE_{hk_j}(\sum_{q=j}^{N}a_Md_q) + HE_{hk_j}(a_Md_j)\}modM\}$

   $= \{HE_{hk_j}(\sum_{q=j}^{N}a_qd_q), \ldots, HE_{hk_j}(\sum_{q=j}^{N}a_Md_q)\}$

   where $hk_j = \sum_{q=j+1}^{N}hk_q$ and $hk_j = hk_j + hk_j$

   $u_j$ sends out $ero_j$ to its parent $u_{j-1}$. The parent does the same thing as $u_j$ to generate data and finally, BS receives an encryption $ero_i$ from $u_i$.

**Step 5. Data recovery** When receiving $ero_1$, BS first computes $hk = \sum_{i=1}^{N}hk_i$ to decrypt $ero_1$ and then uses all $k_i$ to recover $A$ as described before. After that, BS can operate similarly with the first scheme to get the original data $d = \{d_1, d_2, \ldots, d_N\}$. BS does the same thing to recover $d$ for each sub tree. As a result, BS can obtain all the sensed data in WSN.

**VI. Security analysis**

In this section, we analyze the privacy properties of our two schemes. Due to the limited space, we omit some details of the analysis procedure.

**A. Security analysis of the first scheme**

**Compromising-resistance.** In our system model, as data reports are forwarded from leaf nodes toward BS, the adversary can only control compromised nodes to capture the data reports of their downstream nodes. Our first scheme can guarantee that the adversary can only get restricted data reports of its downstream nodes. Generally, we differentiate two types of the compromising attacks: single node compromising and multi node compromising.

1) Single node compromising: In this attack, the adversary compromises only one node and controls it to capture data reports of its downstream nodes. Suppose the adversary selects a sub tree and compromises one node of it, there are three cases:

- Case 1. Compromising a leaf node $u_N$: As $u_N$ is a leaf node and has no child, the adversary controlling $u_N$ cannot capture any data reports of other nodes.

- Case 2. Compromising an intermediate node $u_{N-1}$: By controlling $u_{N-1}$, the adversary can capture the original data report $r_0_{N-1}$ from $u_N$ to infer $d_N$. In this case, we show that the adversary can fix a set which contains $d_N$ with high efficiency. The reason is that we can treat $r_0_N = \{a_1d_N, a_2d_N, \ldots, a_Md_N\}$ as a linear equation system with the number of unknowns larger than the number of linear equations. Therefore, the unique solution of this system does not exist. As a result, $d_N$ is hidden in the set of all the potential solutions. For the adversary, the lower bound of computation complexity to fix this set is $MU$ in the worst case. In real applications, the range $[0, U]$ of the sensed data is often several thousands. As a result, it is efficient for the adversary to find this set.

- Case 3. Compromising an intermediate node $u_j$ ($1 \leq j \leq N-2$): By controlling a node $u_j$, the adversary can capture the compressed data report $r_{0j+1}$ from its child $u_{j+1}$. As $r_{0j+1}$ compresses the data reports of all the $u_j$’s downstream nodes $\{u_j\}_{q=j+1}^{N-1}$ (note that we term the size of $\{u_q\}_{q=j+1}^{N-1}$ as $Z_{j+1}$), the adversary can use $r_{0j+1}$ to infer the knowledge of their sensed data $d = \{d_{1j}, d_{2j}, \ldots, d_N\}$. In this case, we show that $d$ is hidden in a set and it is infeasible for the adversary to fix this set. Specifically, the captured $r_{0j+1}$ can be treated as a linear equation system. The total number of unknowns are $(M+1)Z_{j+1}$ while the total number of linear equations is $M$. Therefore, the unique solution of this
system does not exist. As a result, \( d \) is hidden in the set of all the potential solutions. For the adversary, the lower bound of computation complexity to find this set is \( 2^{(M2j-1-M)+UZ+1} \) in the worst case.

2) Multi node compromising: The adversary can compromise multiple nodes and control them to capture multiple data reports. Suppose the adversary chooses multiple nodes to compromise, there are two cases:

- **Case 1.** The adversary compromises multiple nodes in a single sub tree: For simplicity, we only analyze the situation when two intermediate nodes are compromised. The more complex situations are analogous. Suppose the adversary compromises two intermediate nodes \( u_i \) and \( u_j (N-2>j>i>0) \) and thus, captures two data reports \( ro_{u_i+j} \) and \( ro_{u_j+1} \) from their children. There are two situations. Firstly, the adversary can separately use \( ro_{u_i+1} \) and \( ro_{u_j+1} \) to infer the sensed data \( \{d_{i+1}, d_{i+2}, \ldots, d_N\} \) and \( \{d_{j+1}, d_{j+2}, \ldots, d_N\} \) contained in \( ro_{u_i+1} \) and \( ro_{u_j+1} \) respectively. This situation is similar to case 3 of single node compromising. Secondly, the adversary can jointly use \( ro_{u_i+1} \) and \( ro_{u_j+1} \) to acquire additional advantage. In particular, the adversary can subtract \( ro_{u_i+1} \) by \( ro_{u_j+1} \). The subtraction result \( ro = ro_{u+i+1} - ro_{u+j+1} \) contains the sensed data \( \{d_{i+1}, d_{i+2}, \ldots, d_j\} \). The adversary can thus infer \( \{d_{i+1}, d_{i+2}, \ldots, d_j\} \) from \( ro \). We analyze the adversary’s advantage in this situation with different values of \( i \) and \( j \) and we find that: (1) when \( j=i+1 \), the adversary cannot get additional knowledge, (2) when \( j=i+2 \) this case is similar with the case 2 of single node compromising to get a small set of values which contains \( d_{i+1} \) and (3) when \( j=i+\beta (3\leq\beta\leq N) \), this case is similar with the case 2 of single node compromising where the adversary compromises the intermediate node \( u_{N-i+2} \) in a sub tree.

- **Case 2. The adversary compromises multiple nodes in multiple sub trees:** For simplicity, we also analyze the situation with two nodes. The more complex situations are analogous. Suppose the adversary receives two data reports from the children of the two compromised nodes. However, as the two nodes do not belong to the same sub tree, the adversary cannot subtract the two data reports and thus cannot gain additional advantages by jointly using the two data reports. Furthermore, uniting them into a linear system of equations will lead the numbers of unknowns \( d_{ij} \) increasing two times, providing no advantage for the adversary to infer the sensed data.

Eavesdropping-resistance The resistance of eavesdropping attack is immediate. As all the data reports are transmitted in encryption, the adversary can only acquire encrypted messages by eavesdropping any communication link. On the other hand, it is still interesting to show why the nature of CS technique cannot resist eavesdropping attack. Suppose now all the data reports are transmitted without encryption. If the adversary wants to infer the knowledge of a sensed data \( d_j \) of an intermediate node \( u_j (1\leq j\leq N-2) \), it can eavesdrop the links between \( u_j \leftrightarrow u_{j+1} \) and \( u_j \leftrightarrow u_{j-1} \) to capture two compressed data reports \( ro_{u_j+1} \) and \( ro_{u_j} \) respectively. The adversary can then compute the original data report \( ro_j \) of \( u_j \): \( ro_j = ro_{u_j} - ro_{u_j+1} \). From \( ro_j \), the adversary can efficiently fix a set of values that contains the sensed data \( d_j \) as in the case 2 of single node compromising.

B. Security analysis of the second scheme

**Compromising-resistance** we also differentiate two types of compromising attacks: Single node compromising and Multi node compromising.

1) **Single node compromising:** Suppose the adversary compromises a node \( u_j \) in a sub tree, the adversary can receive an encryption \( er_{ro_{u_j+1}} \) from \( u_{j+1} \). As the adversary does not know the encryption key \( hk \), he cannot decrypt \( er_{ro_{u_j+1}} \).

2) **Multi node compromising:** On the other hand, the adversary can compromise multiple nodes to launch attacks with collusion. By controlling multiple compromised nodes, the adversary can receive a set of encryptions. Contrary to the first scheme, the adversary cannot use the subtract operation on this set no matter whether the compromised nodes are belonging to the same sub tree. This is because the additively homomorphic encryption only supports addition operations on the encryptions. As a result, combining multiple encryptions from multiple nodes cannot help the adversary to decrease the computation complexity. In fact, to decrypt an encryption \( er_{ro_{u_j+1}} \) received from a node \( u_{j+1} \), the adversary has to compromise all the nodes in \( \{u_q\}_{q=j+1}^{N} \) to recover the key \( hk = \sum_{q=j+1}^{N} hq \).

**Eavesdropping-resistance** As all the data reports are under additively homomorphic encryption before forwarding, eavesdropping on any communication link can only reveal encrypted versions of them. The advantage of the adversary to recover the underlying data reports is equal to breaking the additively homomorphic encryption, which is computational infeasible [4].

VII. PERFORMANCE EVALUATION

In this section, we compare the performance of basic encryption scheme, private data aggregation and our schemes.

A. Accuracy

Both basic encryption scheme and our schemes achieve accuracy as individual sensed data can be recovered at BS. The network manager can then use these sensed data to compute any aggregate statistic. By using private data aggregation, however, only an aggregation result can be recovered at BS. The utility of the aggregation result is limited in two folds. Firstly, the aggregation result can only support limited types of aggregate statistics. Secondly, the acquisition of each supported aggregate statistic requires multiple aggregation operations over the whole sensor network, which is energy consuming if the network manager wants to acquire multiple aggregate statistics.
B. Transmission efficiency

We evaluate the transmission efficiency of the three schemes according to two criteria: energy balance and overall energy consumption. For energy balance, we compare the data sent by different nodes in each of the three schemes. For overall energy consumption, we use the bits of forwarding data to measure the performance of the three schemes. Note that in our system model, the data collection is performed in each sub tree independently. As a result, we will also compare the performance of the above schemes in sub tree basis.

Energy balance The energy balances of the three schemes are analyzed as follows:
• Basic encryption scheme: In this scheme, each node needs to forward the encryptions of all the sensed data generated from its downstream nodes as well as itself. In particular, the node \( u_j \) (\( 1 \leq j \leq N \)) needs to forward \((N-j+1)\) encryptions. The result reflects a large amount of communication load among sensors with different indexes. In the extreme case, the leaf node \( u_N \) only forwards one encryption and the root node \( u_1 \) needs to forward \( N \) encryptions. As a result, the nodes closer to BS have to send significantly larger amounts of data than downstream nodes.
• Private data aggregation: In private data aggregation, the encrypted sensed data are aggregated at each intermediate node. As a result, each node sends an encryption no matter its location. As opposed to the basic encryption scheme, the energy consumption per node is equal.
• Our schemes: In both of our two schemes, data reports can be compressed in each intermediate node and each node only sends an encrypted compressed data report no matter its location. As a result, the transmitted data per node is equal. In the first scheme, each node needs to transmit one symmetric encryption and in the second scheme, each node needs to transmit \( M \) homomorphic encryptions.

Overall energy consumption We assume that a sensed data \( d_j \) is a \( r' \) length bit string and a random number \( a_{ij} \) is a \( r \) length bit string. We choose the symmetric encryption proposed in [10] to instantiate the basic encryption scheme and our first scheme. The overall energy consumption of the three schemes are analyzed as follows:
• Basic encryption scheme: In this scheme, a symmetric encryption of a sensed data generated at a node will be forwarded to BS along the path from that node to the root. As a result, the symmetric encryption generated at \( u_j \) (\( 1 \leq j \leq N \)) is forwarded \( j \) hops. The total bits of forwarded data over the whole sub tree is thus \( N(N+1)r' \).
• Private data aggregation: In this scheme a node \( u_j \) (\( 1 \leq j \leq N \)) forwards one aggregated data to its parent \( u_{j-1} \). As a result, the bits of forwarded data over the whole sub tree is \( N(\log_2 N+r') \).
• Our schemes: In our first scheme, each node sends a symmetric encryption. As a result, the bits of forwarded data over the whole sub tree is \( N(M(\log_2 N+r')+r') \). In our second scheme, each node sends a homomorphic encryption. As a result, the bits of forwarded data over the whole sub tree is \( NM(\log_2 N+r'+r) \).

C. Simulation result

We further compare our schemes with basic encryption and private data aggregation through simulation. The accuracy of our schemes is guaranteed by the CS technique which shows that the compressed data can be recovered with a precision close to one if \( M \) is large enough. In real application, the value of \( M \) is based on the physical characteristic of the sensing area and the kind of collected data. In this paper, we select the value of \( M \) based on [13], which shows that \( M=0.05N \) (recall \( N \) is the size of a sub tree) is large enough to recover the temperature data of ocean sensed by sensors [14] with a precision over 98%. We also set \( r'=16 \) and \( r=32 \). The simulation is performed on two typical topologies: chain and tree. Chain-type sensor network consists of a single route and is often deployed along narrow areas like rivers, streets or tunnels. We formalize this topology as a collection tree with \( Q=1 \). Tree-type sensor network consists of multiple routes and is often deployed in wide areas such as round areas or square areas. We formalize this topology as a collection tree with \( Q>1 \). In this simulation, we set \( Q=10 \).

Energy balance: To evaluate the energy balance, we fix the parameter \( N \) of chain topology to 500 and tree topology to 100 and analyze the number of forwarded data items at nodes in different hops. Fig. 2 (a) and (b) show the energy balance of the three schemes on the chain topology and tree topology at different hops of a sub tree. We can see that the basic encryption suffers from energy unbalance since
the forwarding number of data items increases linearly as the hop number decrease. On the other hand, the energy is fairly dispensed to each node in our two schemes and private data aggregation.

**Overall energy consumption:** To evaluate the overall energy consumption, we vary the parameter $N$ of the chain topology from 100 to 500 and tree topology from 100 to 200. As the performance of our two schemes are same, we only use the second scheme for comparison. Fig. 2 (c) and (d) show the overall energy consumption of the three schemes on the chain topology and tree topology respectively as $N$ changes. We can see that basic encryption scheme consumes much higher energy than our schemes and private data aggregation. Also, the overall energy consumption of our two schemes is slightly higher than private data aggregation. This coincides with the intuition that the private data aggregation sacrifices data accuracy. Further, our second scheme consumes higher energy than our first one. This is because the length of additive homomorphic encryption is longer than the symmetric encryption.

**VIII. Conclusion**

In this paper, we thoroughly analyze the requirements of data collection in wireless sensor networks including accuracy, privacy and efficiency. We then propose two private data collection protocol to fulfill these requirements. We conduct comprehensive analysis and prove the correctness and efficiency of the proposed protocols.

**IX. Acknowledgment**

We would like to thank the anonymous reviewers for their helpful and valuable comments. This work is supported in part by Nanyang Technological University NAP Grant M4080738.020.

**References**


